Project Planning And Tracking With Self-directed Teams

Noopur Davis

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213-3890

Trademarks and Service Marks

The following are service marks of Carnegie Mellon University.

- Capability Maturity Model Integration℠
- CMMI℠
- Team Software Process℠
- TSP℠
- Personal Software Process℠
- PSP℠

The following are registered trademarks of Carnegie Mellon University.

- Capability Maturity Model®
- CMM®

Outline

Importance of project planning and tracking

Self-directed teams and the TSP

Project planning with the TSP launch

Project monitoring and control with the TSP

Project management at capability level 5

Planning And Tracking

Planning and tracking are integral to model-based software process improvement.

CMM

- Software Project Planning (SPP)
- Software Project Tracking and Oversight (SPTO)
- Integrated Software Management (ISM)
- Pervasive in all other key process areas

CMMI

- Project Planning (PP)
- Project Monitoring and Control (PMC)
- Integrated Project Management (IPM)
- Risk Management (RM)
- Quantitative Project Management (QPM)
- Generic goals in all process areas (PAs)

Commitment

The main reason that plans are needed is to make a commitment.

The main reason that plans are tracked and adjusted is to ensure that the commitments can be met.

Commitments must be

- freely assumed
- visible
- managed

Self-directed Teams and the TSP

The TSP uses a self-directed team approach to software development. It is based on the following principles.

- The engineers know the most about the job and can make the best plans.
- When engineers plan their own work, they are committed to the plan.
- Precise project tracking requires detailed plans and accurate data.
- Only the people who are doing the work can collect precise and accurate data.
- Plans change constantly. People who do the work are most able to dynamically replan and track the work.
- To minimize cycle time, the engineers must balance their own workloads.
- To maximize productivity, focus first on quality.

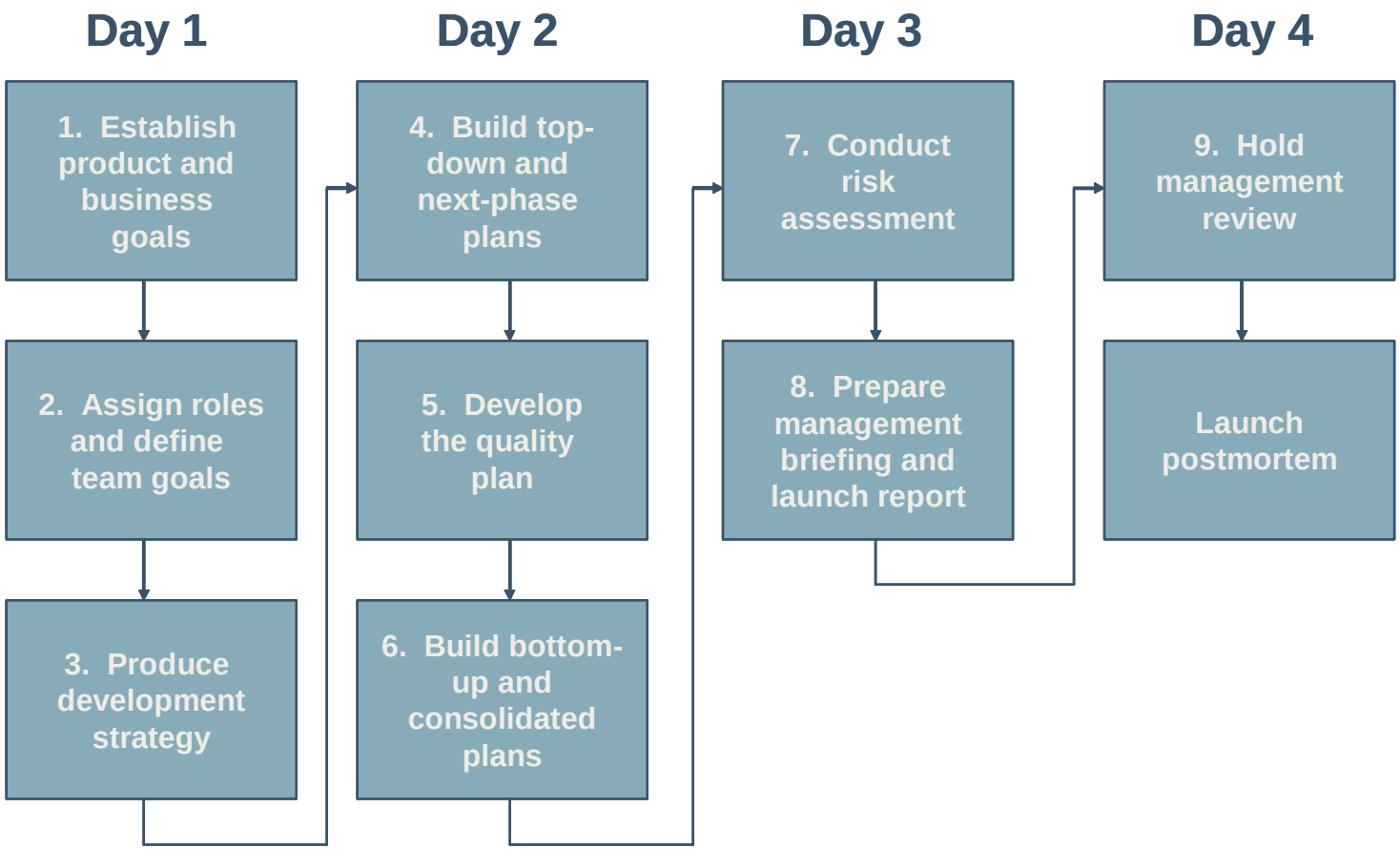
The Team Software Process

Teams using the TSP
- develop their own plans
- make their own commitments
- track and manage their own work

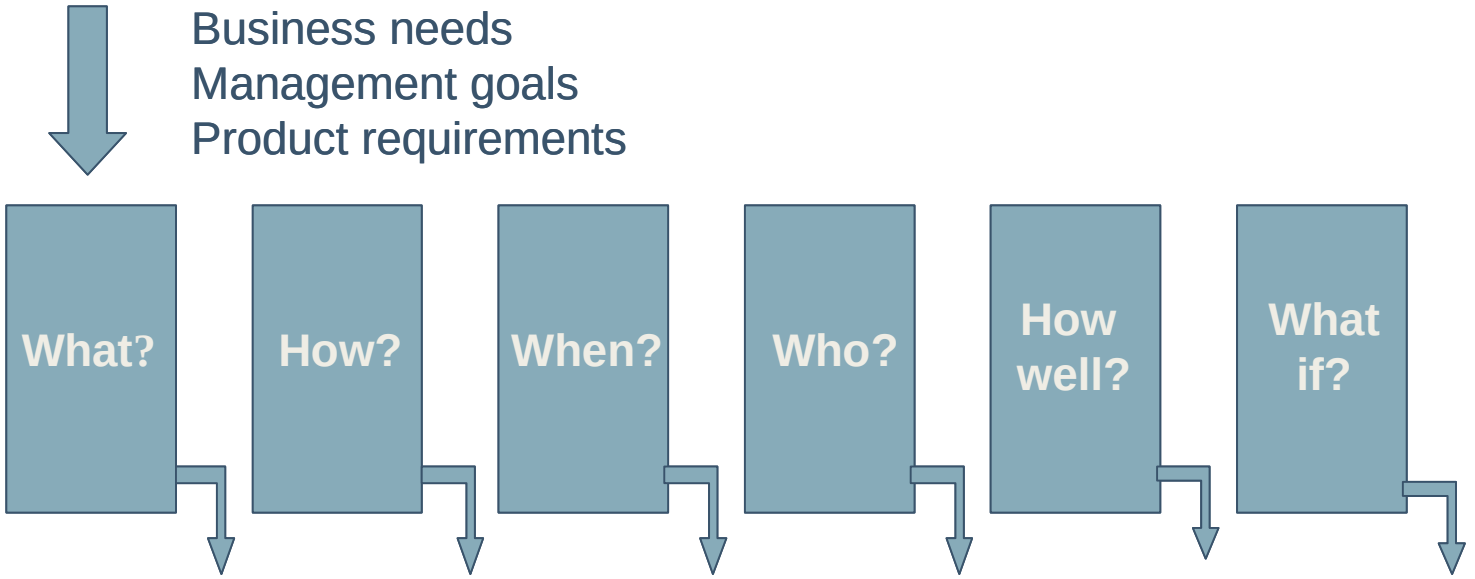
Typical performance
- cost and schedule performance within +/- 10%
- product quality between 0.0 and 0.1 defects/KLOC
- productivity and cycle-time improvements of up to 50%
- test costs reduced by 5 to 10 times

Every TSP project starts with a launch. The launch
- is a defined planning process
- facilitates team-building
- is led by a trained team coach
- follows PSP and TSP training

Project Planning and the TSP Launch



The TSP Launch Products



- Team goals
- Conceptual design
- Planned products
- Size estimates

- Team strategy
- Team defined process

- Task plan
- Schedule plan
- Earned-value plan

- Team roles
- Task plans
- Earned-value plan

- Quality plan

- Risk evaluation
- Alternate plans

TSP/PSP Measurement Framework

"Quality without numbers is just talk."

TSP and PSP use four base measures.

- size
- time in phase
- defects injected and removed by phase
- task-completion date

Both plan and actual data are gathered and recorded.

- Actual data are used for monitoring and controlling the plan.
- Individual and team data are archived to provide a repository of historical data for future use.

Many other measures are derived from the base measures.

Sample of Derived Measures

Estimation accuracy (size/time)
Prediction intervals (size/time)
Time-in-phase distribution
Defect injection distribution
Defect removal distribution
Productivity
Percent reuse
Cost performance index
Planned value
Earned value
Predicted earned value

Defect density
Defect density by phase
Defect removal rate by phase
Defect removal leverage
Review rates
Process yield
Phase yield
Failure cost of quality
Appraisal cost of quality
Appraisal/Failure COQ ratio
Percent defect free
Defect removal profiles
Quality profile
Quality profile index

Project Monitoring And Control

The next few slides show how the measurement framework supports project monitoring and control.

All data shown is available

- for each team member
- for the entire project
- by phase
- by week
- by product component
- by task

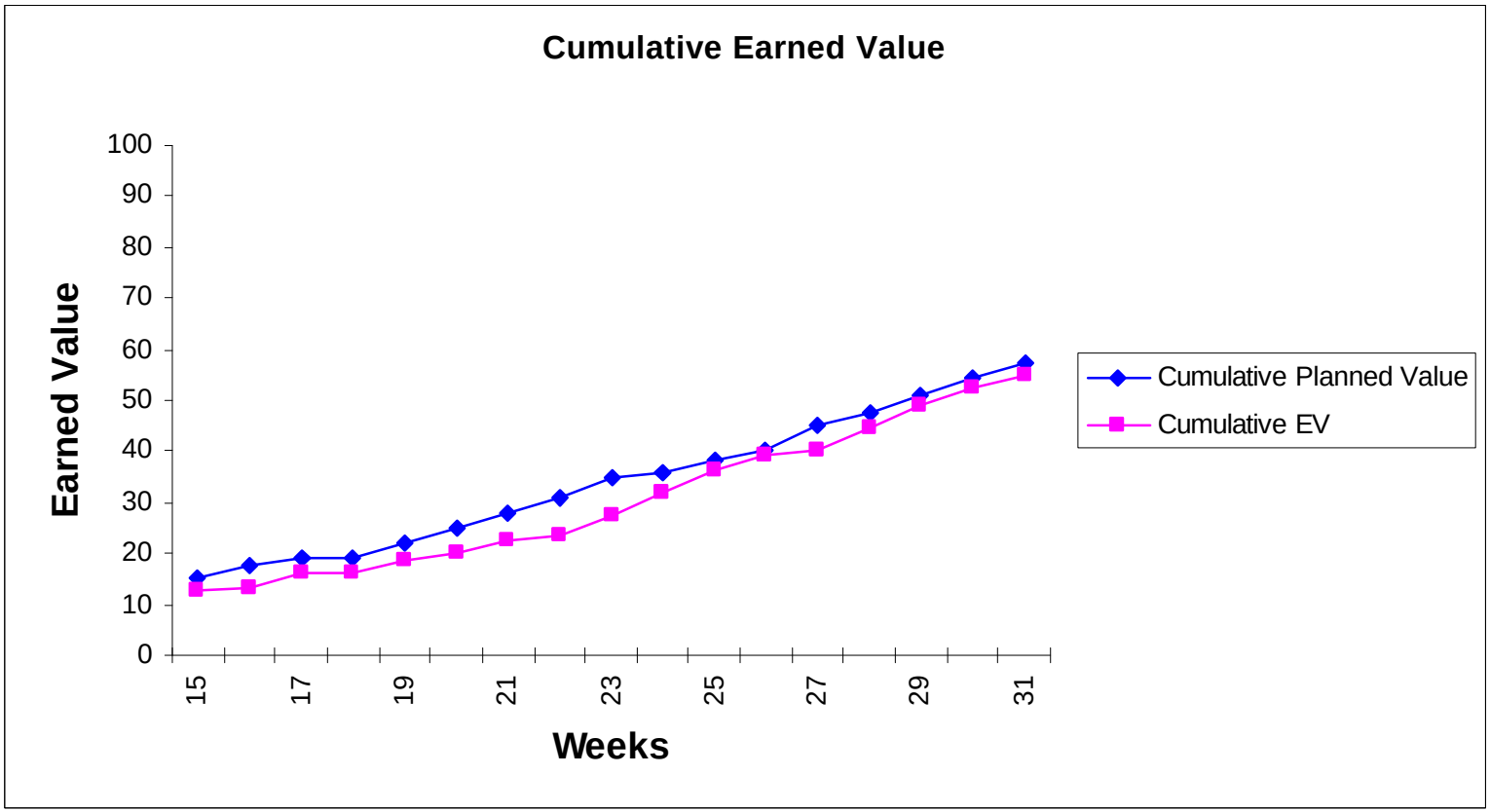
The data is from an actual TSP project.

Project Status – Week 31

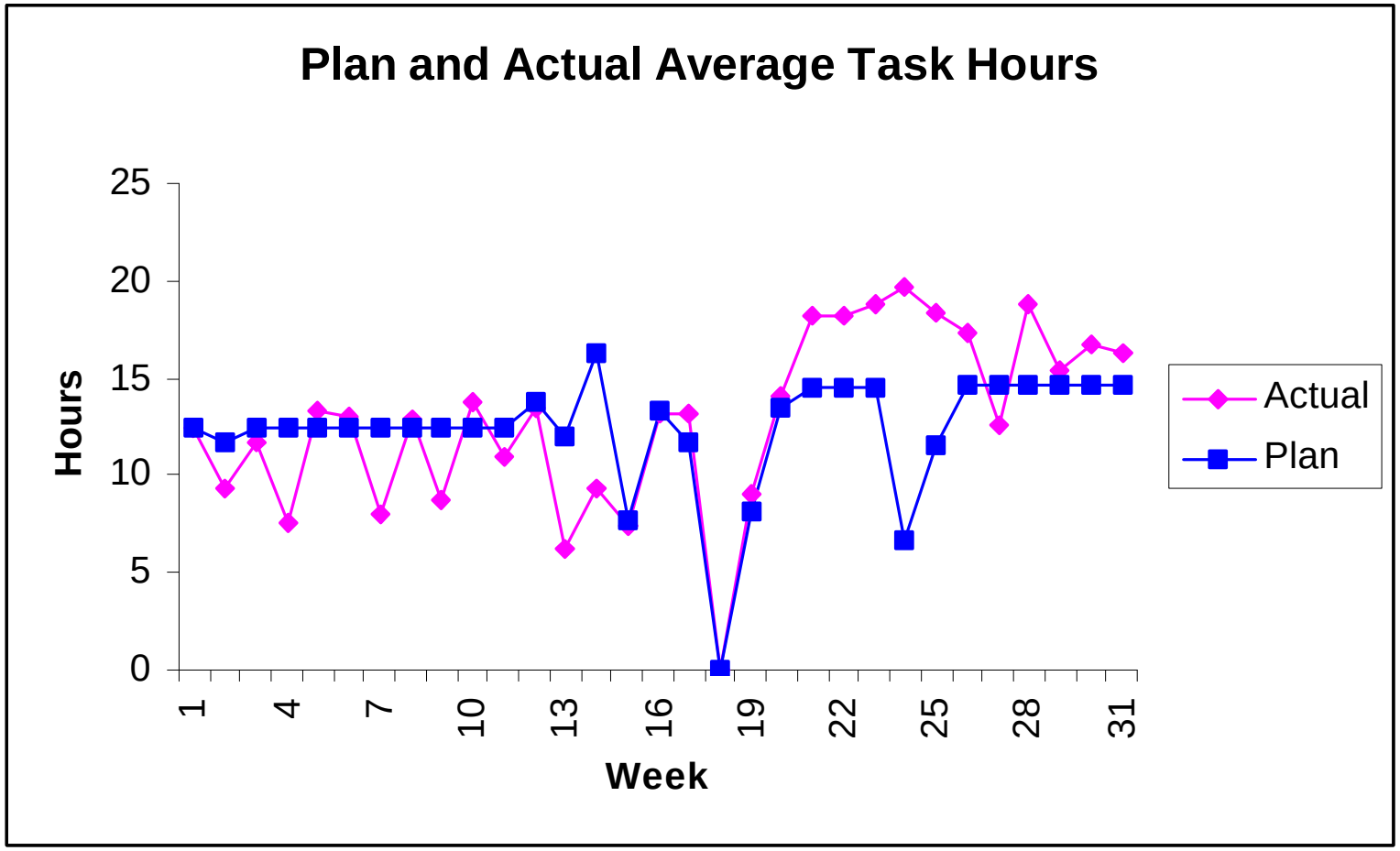
Weekly Data	Plan	Actual	Plan/ Actual
Project hours for this week	102.5	113.1	0.91
Project hours this cycle to date	1902.7	2359.5	0.81
Earned value for this week	2.9	2.5	1.17
Earned value this cycle to date	59.3	58.3	1.02
To-date hours for tasks completed	1835.4	2196.8	0.84

- To-date EV is 2% below plan (59.3 vs. 58.3). Is this a good status indicator?
- 162.7 hours (2359.5 - 2196.8) have been spent on incomplete tasks. Why?
- If these uncompleted tasks did not contribute EV, the job would take 22 more weeks to complete. Is this correct?
- Effort has been underestimated by 16%. Is this a trend?
- The team has compensated by putting in more hours (19%). Can this be sustained?

Earned Value – Week 31



Plan vs. Actual Hours – Week 31



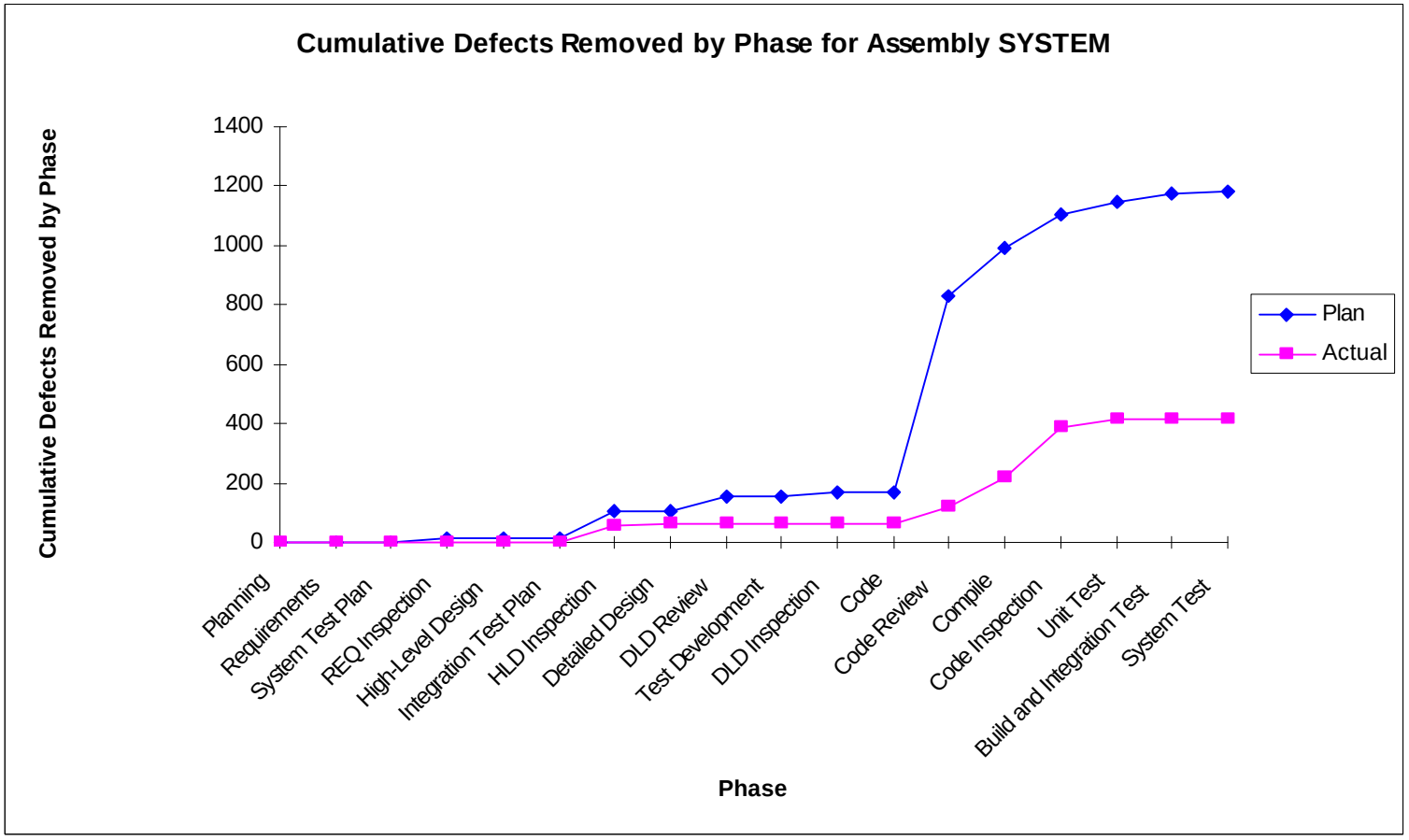
Time in Phase – Week 31

Time in Phase (hours)	Plan	Actual	Actual%
Management and Miscellaneous	16.0	14.4	0.6%
Launch and Strategy	0.0	0.0	0.0%
Planning	47.5	74.2	3.1%
Requirements	80.0	71.2	3.0%
System Test Plan	0.0	0.0	0.0%
REQ Inspection	13.0	25.6	1.1%
High-Level Design	***499.8***	***555.2***	***23.4%***
Integration Test Plan	8.0	27.7	1.2%
HLD Inspection	***405.1***	***170.9***	***7.2%***
Detailed Design	40.2	25.4	1.1%
DLD Review	40.2	4.6	0.2%
Test Development	***168.6***	***121.3***	***5.1%***
DLD Inspection	0.0	0.0	0.0%
Code	468.9	377.5	15.9%
Code Review	***175.1***	***90.7***	***3.8%***
Compile	***136.2***	***167.8***	***7.1%***
Code Inspection	***461.4***	***278.3***	***11.7%***
Unit Test	***415.2***	***203.9***	***8.6%***
Build and Integration Test	111.0	144.8	6.1%
System Test	23.0	0.0	0.0%
Documentation	40.0	0.0	0.0%
Postmortem	0.0	19.4	0.8%
Total	3149.2	2372.8	100.0%

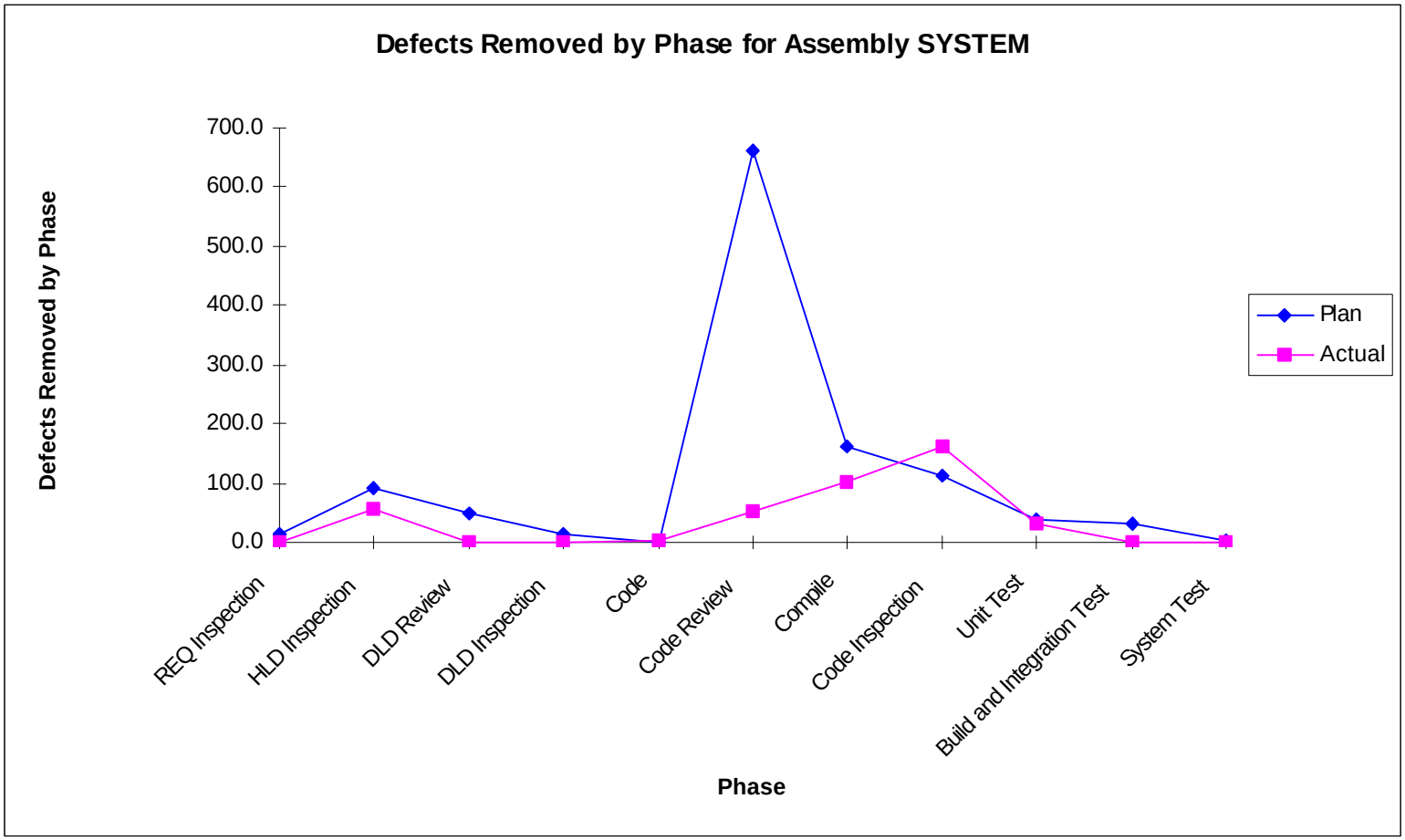
Size Summary – Week 31

Program Size	Plan	Actual
Total Requirements Pages (SRS)	0	0
Total HLD Pages (SDS)		
Total Detailed Design Lines		
Base LOC (B)	83726	34376
Deleted LOC (D)	41	2581
Modified LOC (M)	1741	1985
Added LOC (A)	9899	3294
Reused LOC (R)	2550	28757
New and Changed LOC (N)	11640	5279
Total LOC (T)	96134	63846
Total New Reuse LOC	5304	0
Estimated Object LOC (E)		
Upper Prediction Interval (70%)		
Lower Prediction Interval (70%)		

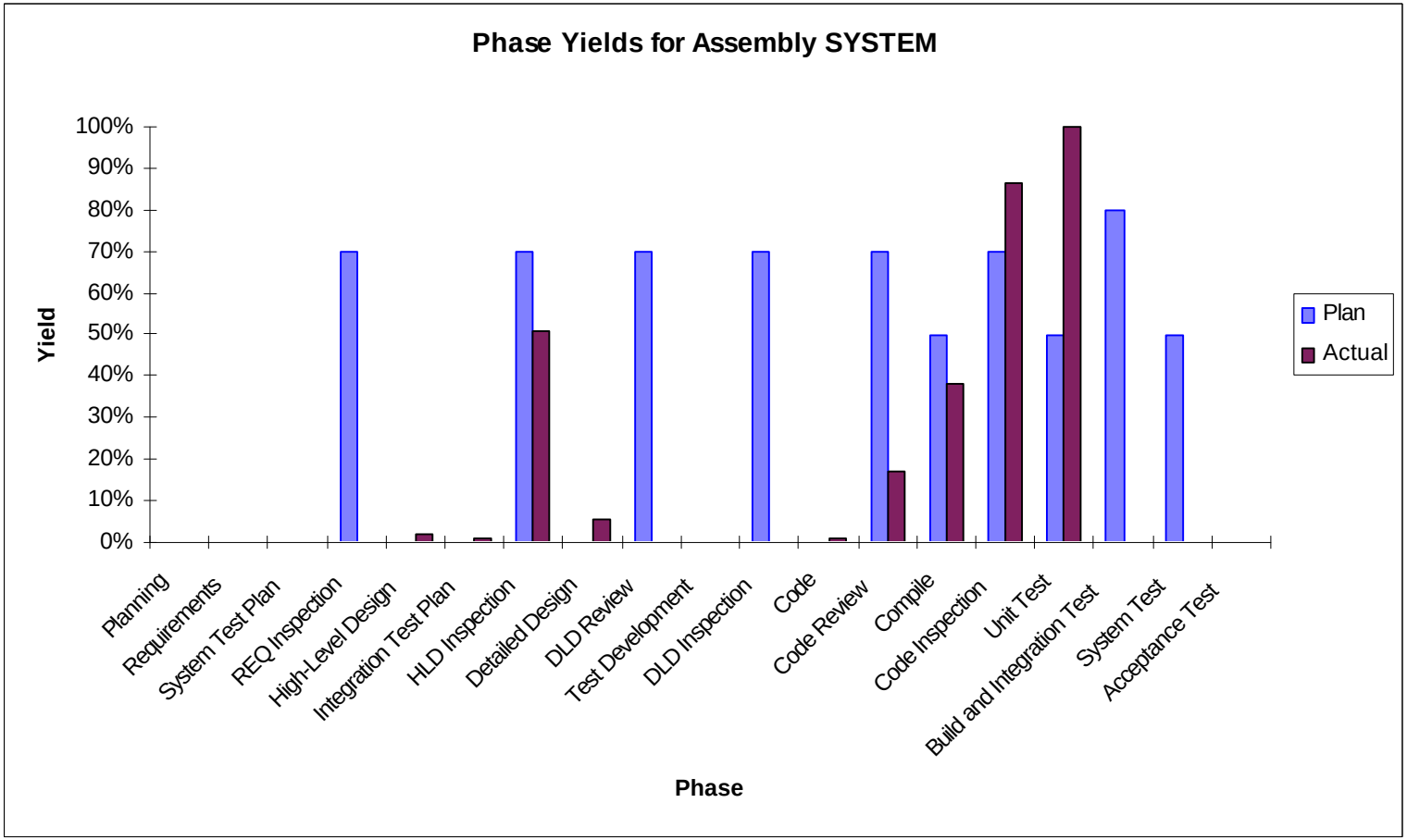
Cumulative Defects Removed – Week 31



Defects Removed by Phase – Week 31



Phase Yields – Week 31



Overall Status – Week 31

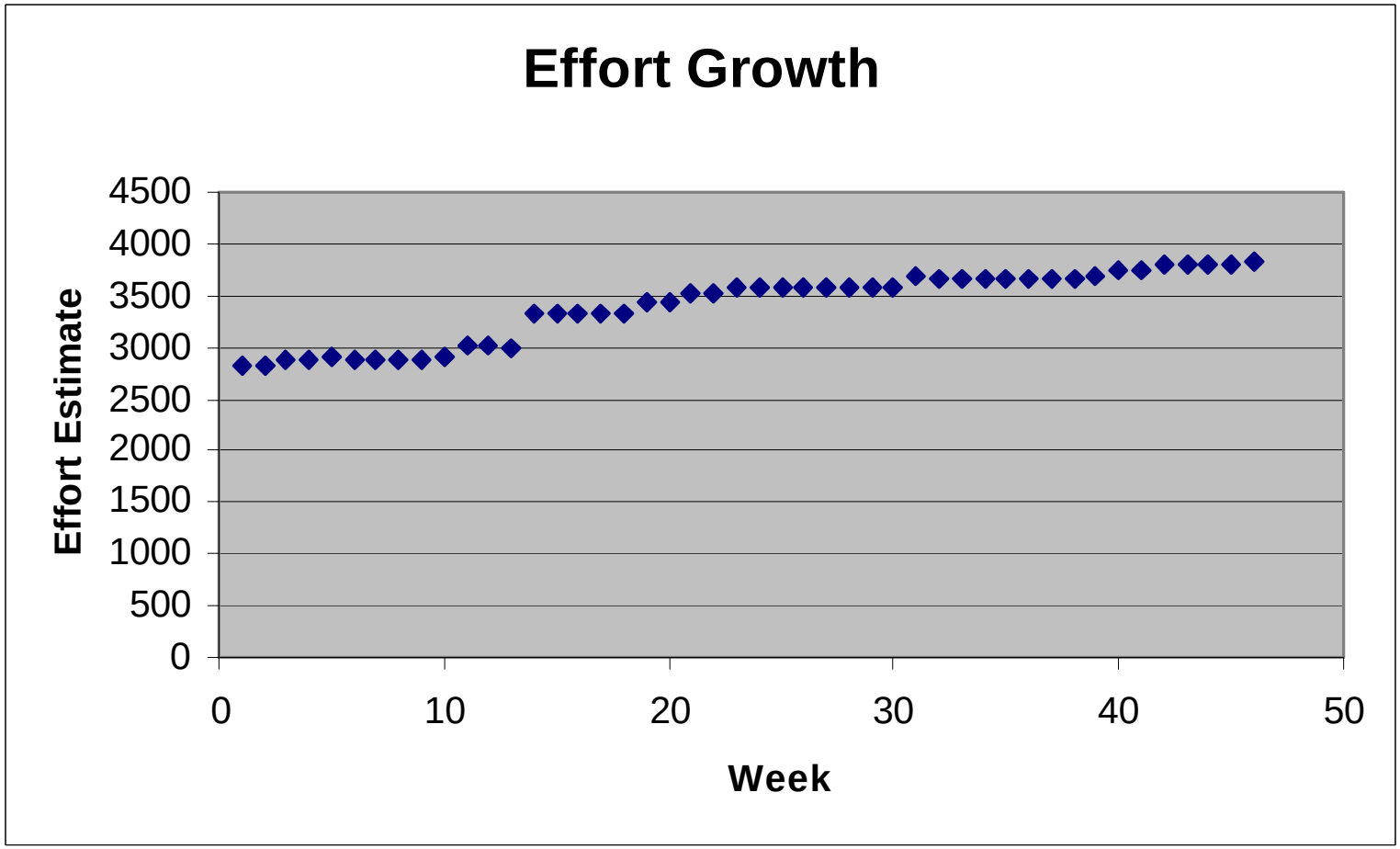
Overall, the project schedule appears to be on track.

It is too early to predict quality. Poor quality will result in schedule delays.

Some positive indicators of quality can be seen. The team
- has put effort into the design
- is conducting design inspections
- is conducting code inspections
- is putting effort in test development
- has found and fixed 423 defects through week 31

Some contrary indicators are insufficient effort in
- personal design reviews
- personal code reviews

The Importance Of Replanning



Capability Level 2 and Beyond

TSP teams use statistical techniques for quantitative process management, including

- flow charts
- scatter diagrams
- histograms
- Pareto analyses
- run (trend) charts
- correlation and linear regression
- prediction intervals

The four basic measures, the derived measures, and the quantitative management techniques can be used to go beyond capability level 2 (CL2).

Size Estimation

We will examine the size measure and its use in the size-estimation subprocess from CL2 through CL5.

At CL2, the goals of the size measure are to

- consistently measure size
- normalize time and defect data
- derive effort estimates

Size Estimation at CL2 -1

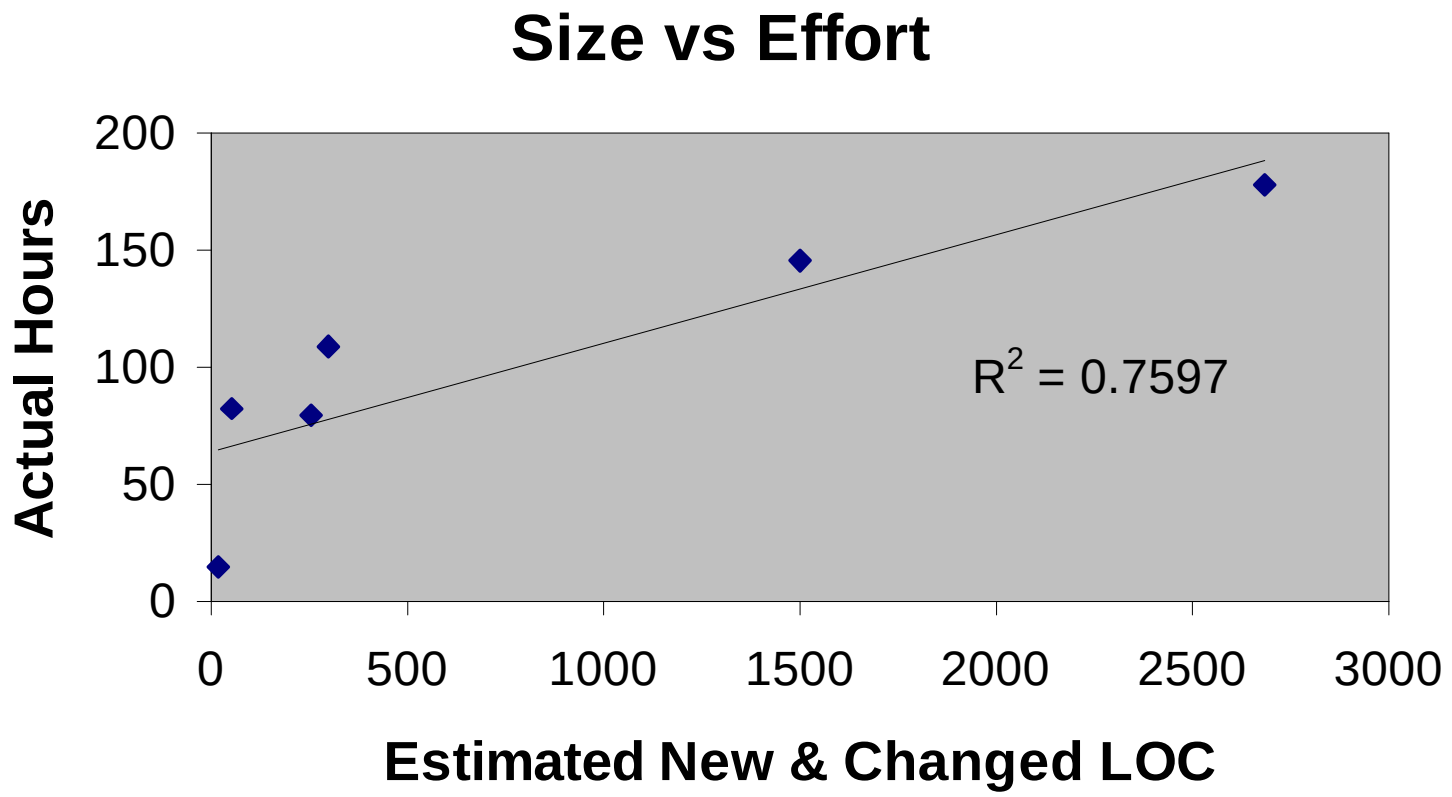
Selecting a measure is simple. An operational definition of a measure is needed.

If lines of code (LOC) is chosen as the size measure for code, LOC must

- be consistently measured
  - coding standards
  - counting standards
- normalize time and defect data
  - defect density (defects per thousand lines of code)
  - productivity (lines of code per hour)
- correlate to effort

Size Estimation at CL2 -2

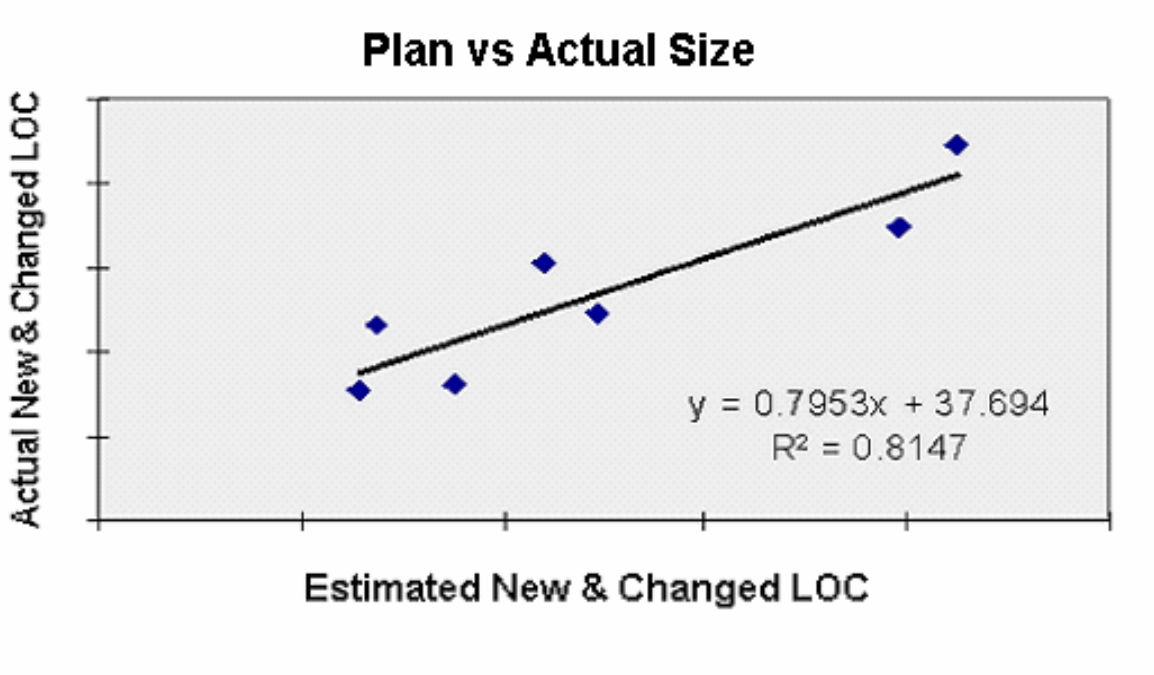
Size measure must correlate to effort.



Size Estimation at CL3

Size estimation can be improved by

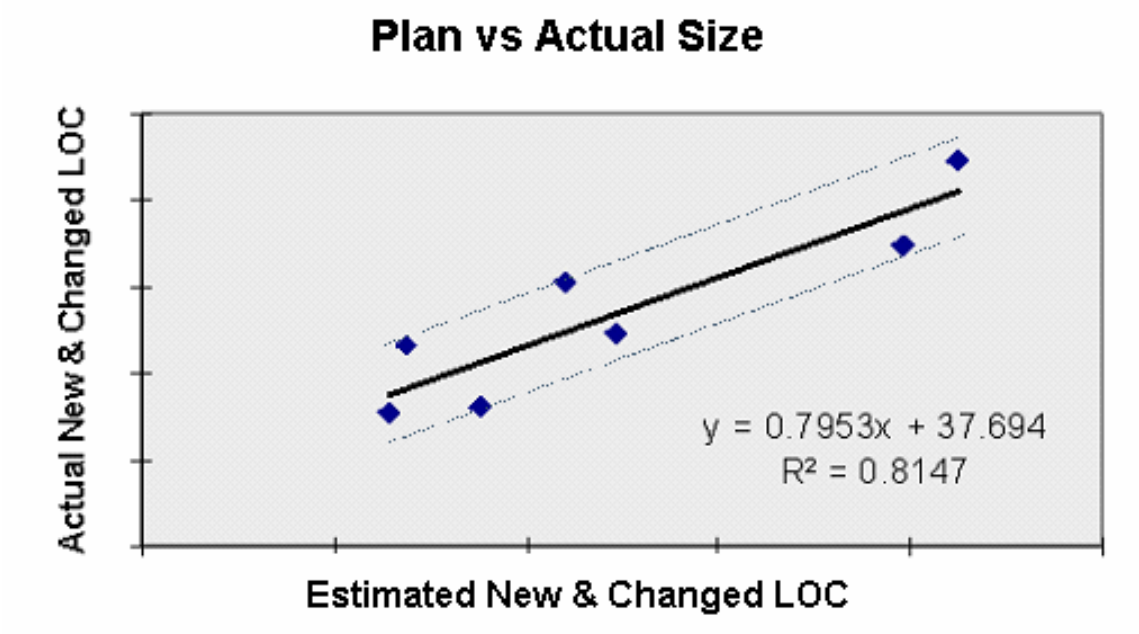
- using defined estimation procedures
- collecting historical data at individual and project level
- adjusting estimates with historical estimation error (linear regression)



Size Estimation at CL4

Understand

- size estimation capability (prediction intervals)
- special causes of variation (data outside the 70% and 90% prediction intervals

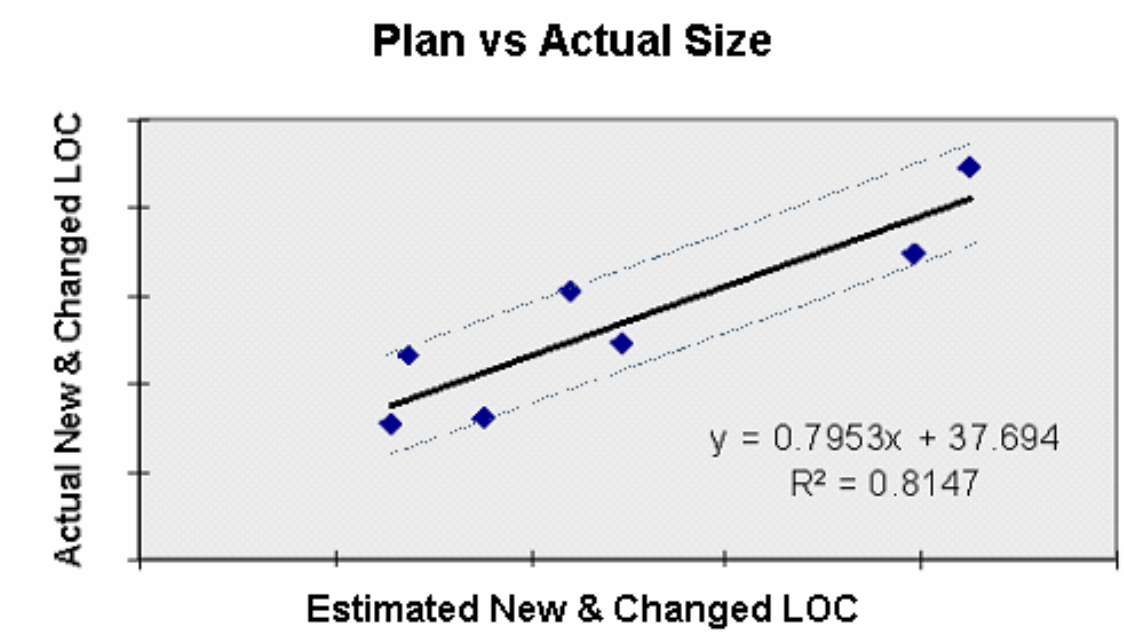


Size Estimation at CL5

Optimize size estimation and understand common causes of variation.

Improve

- linear regression parameters, $?_0$ and $?_1$ (intercept and slope)
- range for prediction intervals (narrower range)



Conclusion

Project planning and tracking is key to model-based process improvement.

Plans are needed to make commitments.

The people who do the work should make their own commitments.

Data should be gathered by people as they do their work.

Data can help people to track and adjust their work to meet their commitments.

Teams can produce extraordinary results when they

- develop their own plans
- make their own commitments
- manage themselves

For More Information

Visit the Software Engineering Institute web site at www.sei.cmu.edu

Visit the TSP web site at www.sei.cmu.edu/tsp

Contact SEI Customer Relations at 412-268-5800

Contact Noopur Davis at nd@sei.cmu.edu